

Fig. 1

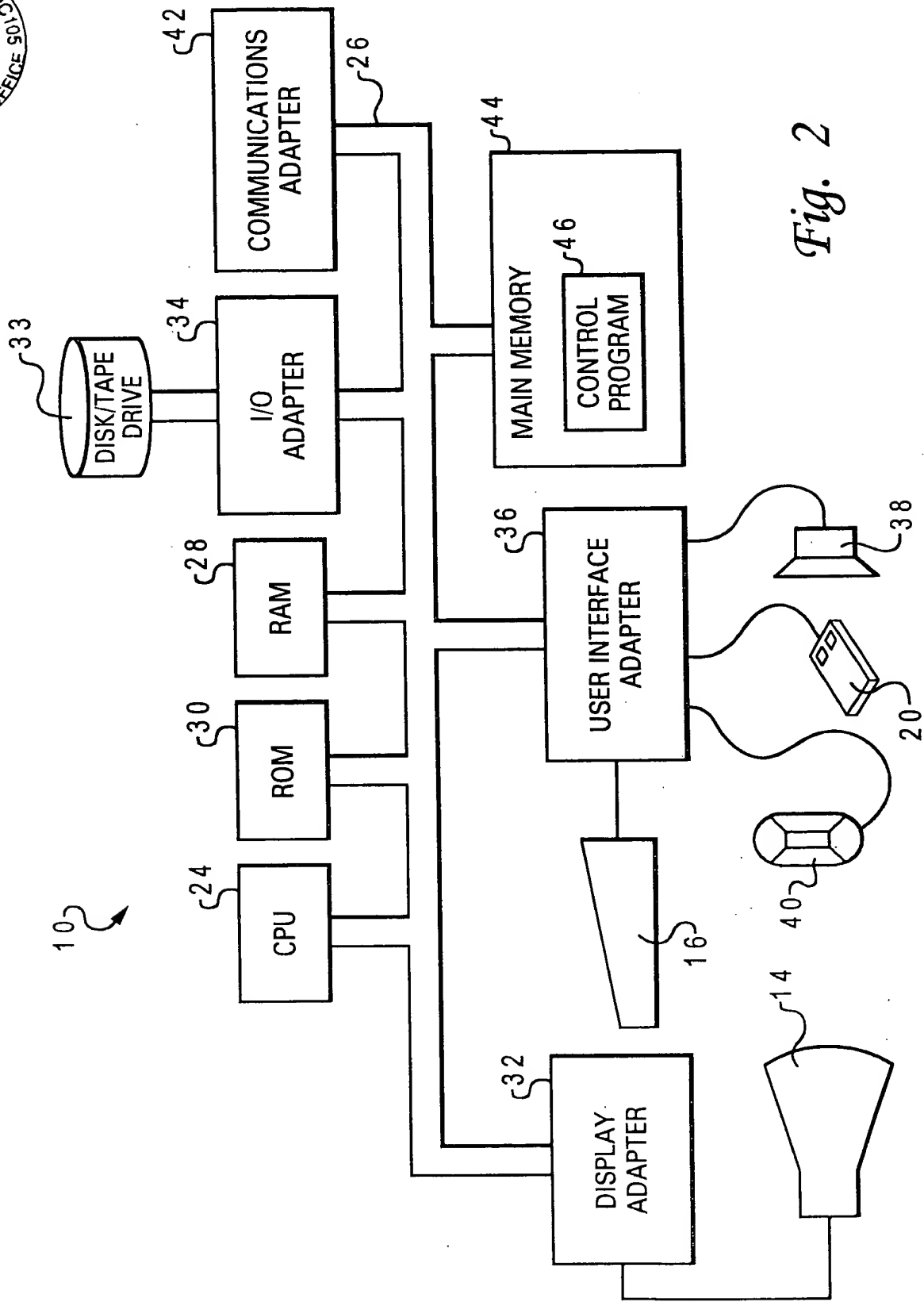


Fig. 2

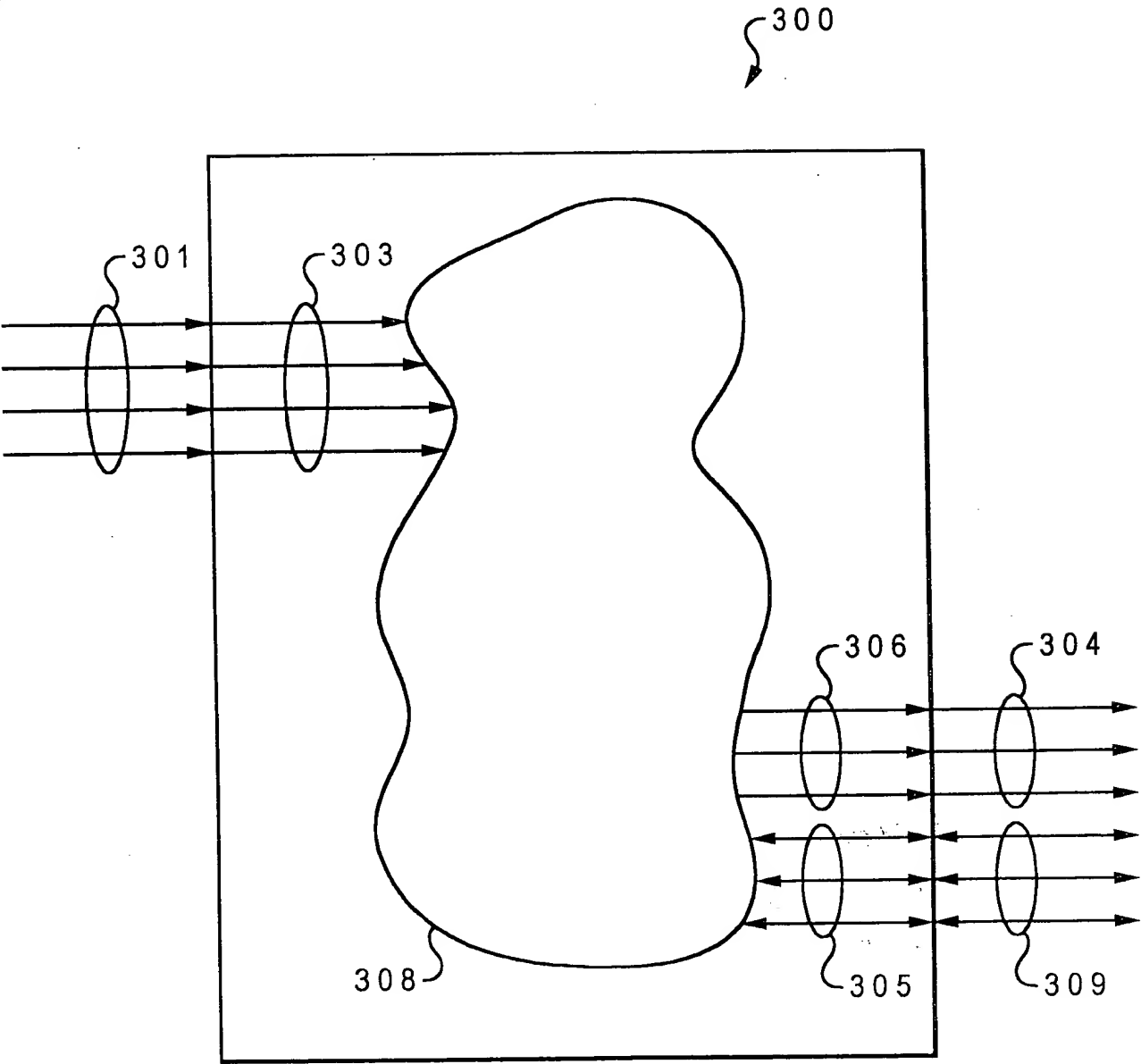


Fig. 3A



4/26

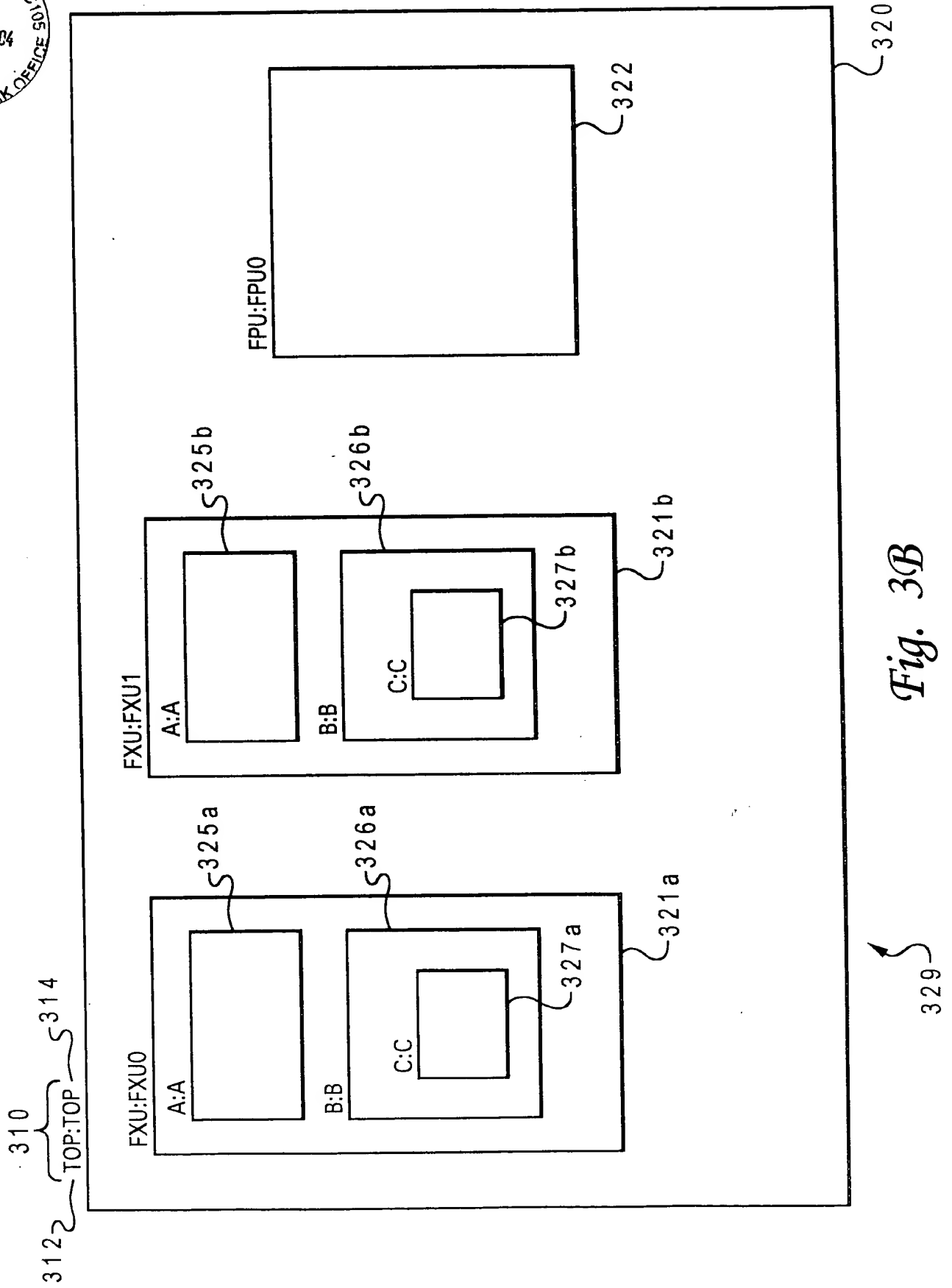


Fig. 3B

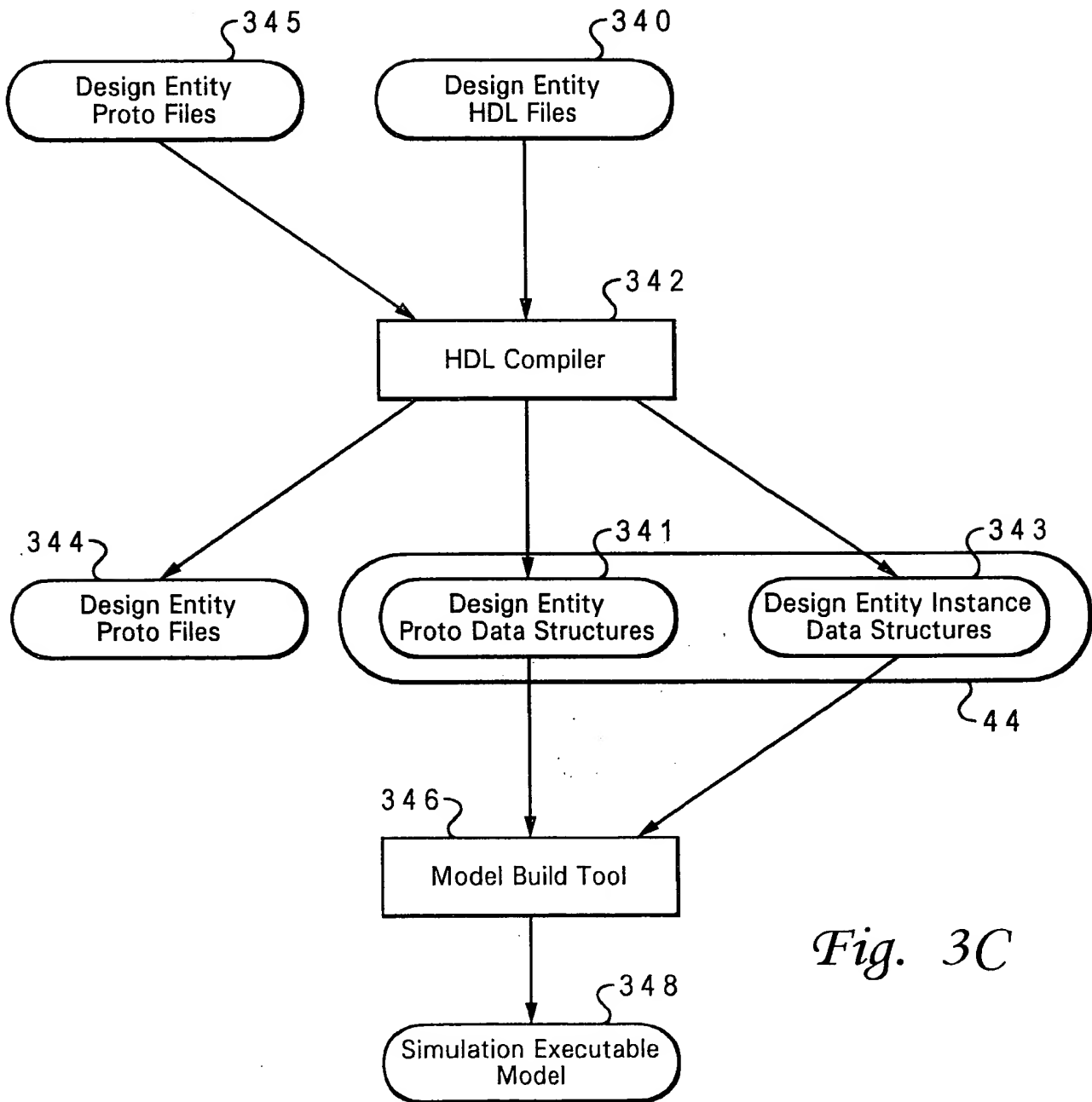
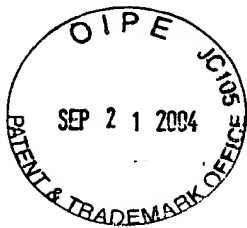


Fig. 3C



6/26

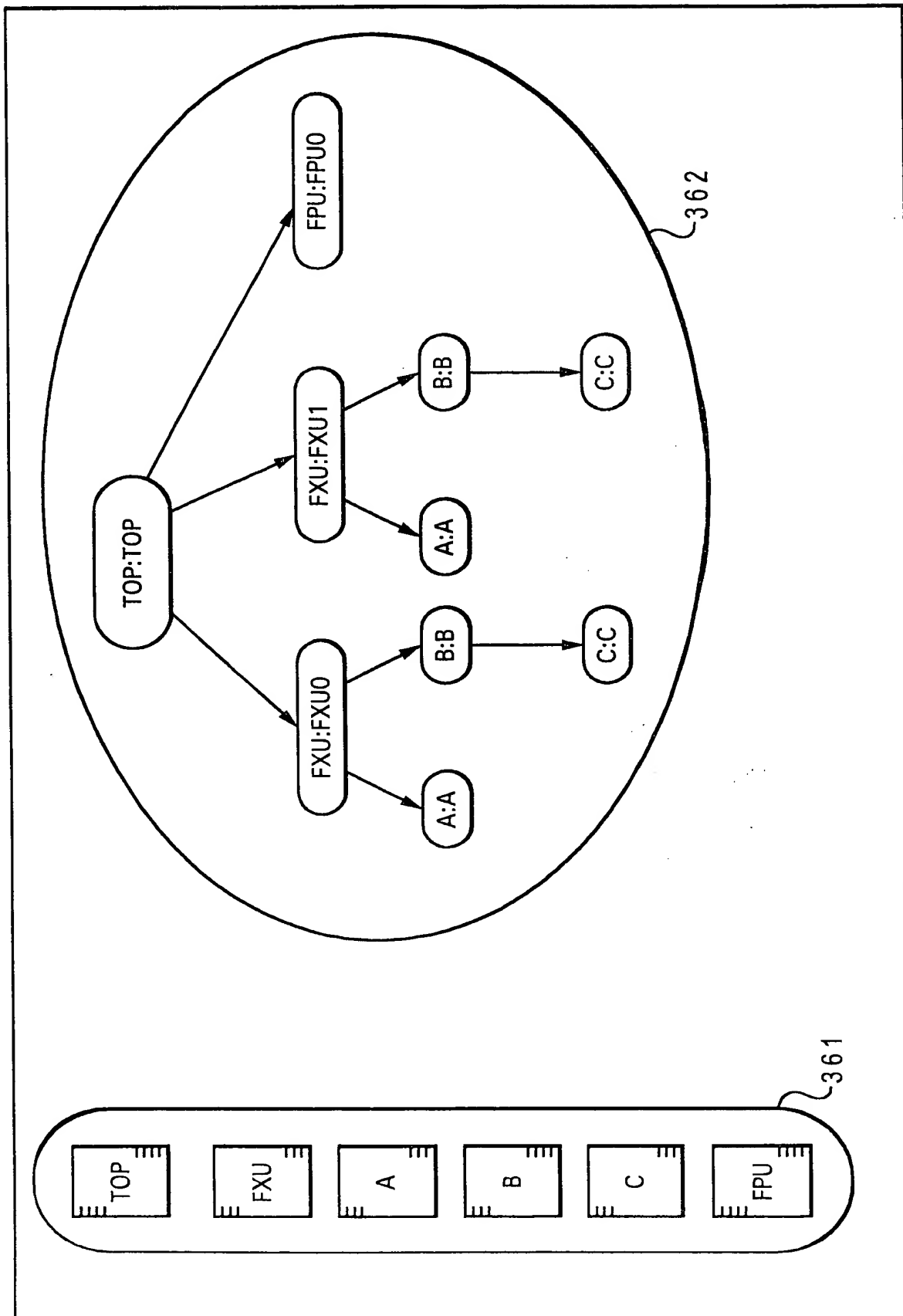


Fig. 3D

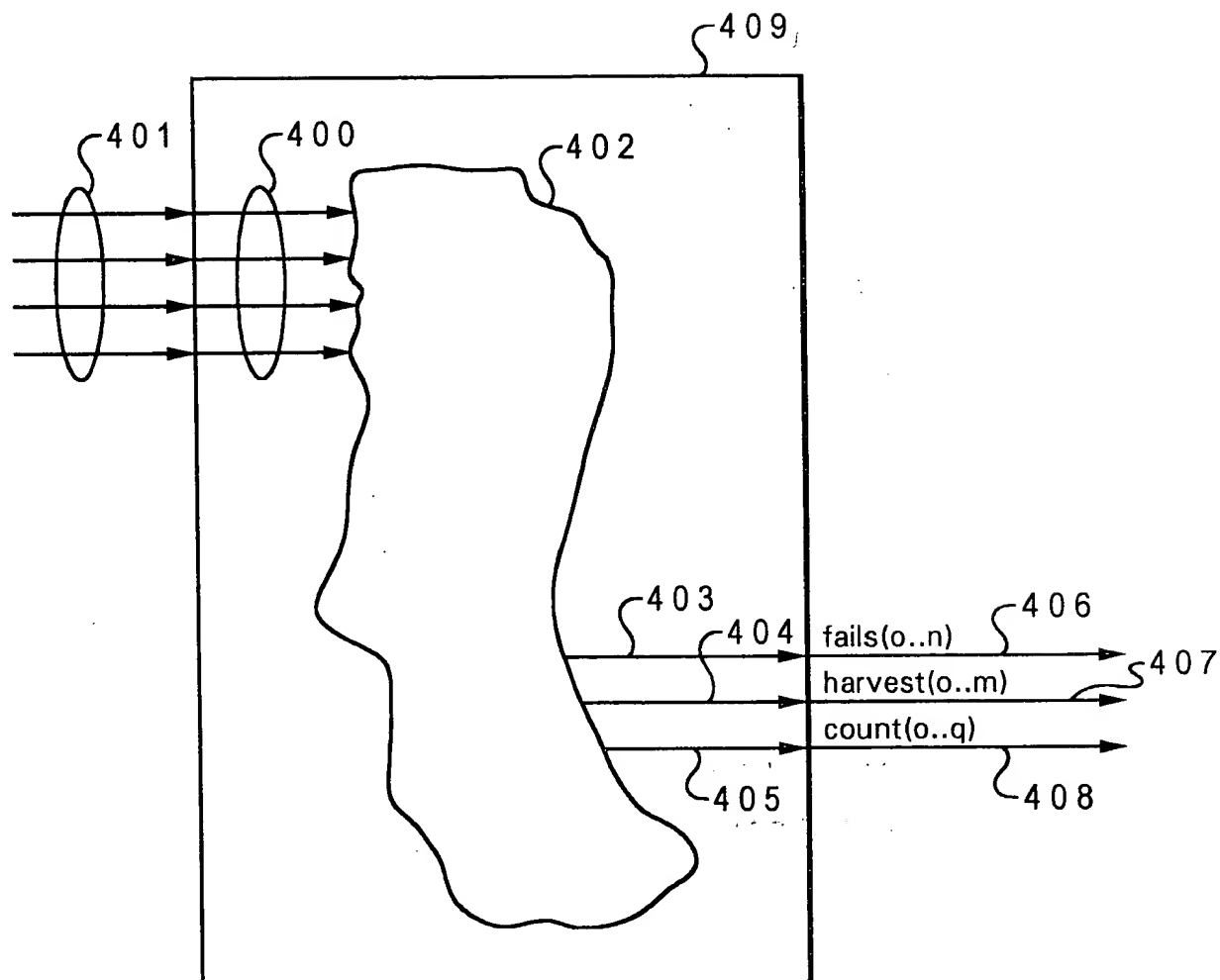


Fig. 4A



8/26

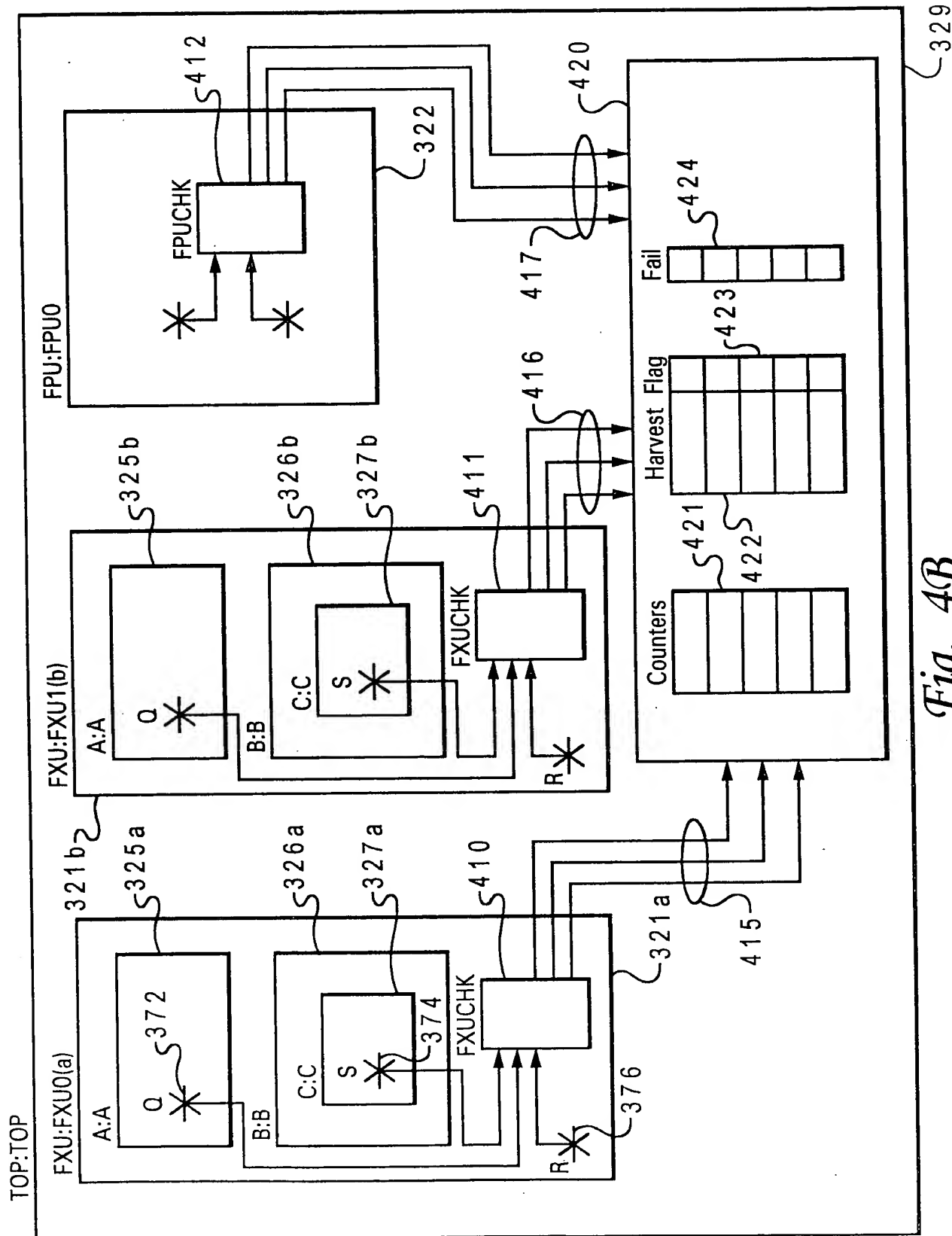


Fig. 4B

9/26



ENTITY FXUCHK IS

```

PORT(  S_IN      :    IN std_ulogic;
        Q_IN      :    IN std_ulogic;
        R_IN      :    IN std_ulogic;
        clock     :    IN std_ulogic;
        fails     :    OUT std_ulogic_vector(0 to 1);
        counts    :    OUT std_ulogic_vector(0 to 2);
        harvests  :    OUT std_ulogic_vector(0 to 1);
);

```

4 5 0

```

4 5 2 { --!! BEGIN
      --!! Design Entity: FXU;

```

```

4 5 3 { --!! Inputs
      --!! S_IN      =>    B.C.S;
      --!! Q_IN      =>    A.Q;
      --!! R_IN      =>    R;
      --!! CLOCK     =>    clock;
      --!! End Inputs

```

```

4 5 4 { --!! Fail Outputs;
      --!! 0 : "Fail message for failure event 0";
      --!! 1 : "Fail message for failure event 1";
      --!! End Fail Outputs;

```

```

4 5 5 { --!! Count Outputs;
      --!! 0 : <event0> clock;
      --!! 1 : <event1> clock;
      --!! 2 : <event2> clock;
      --!! End Count Outputs;

```

```

4 5 6 { --!! Harvest Outputs;
      --!! 0 : "Message for harvest event 0";
      --!! 1 : "Message for harvest event 1";
      --!! End Harvest Outputs;

```

```

4 5 7 { --!! End;

```

4 5 1

4 4 0

ARCHITECTURE example of FXUCHK IS

BEGIN

... HDL code for entity body section ...

END;

4 5 8

Fig. 4C

10/26

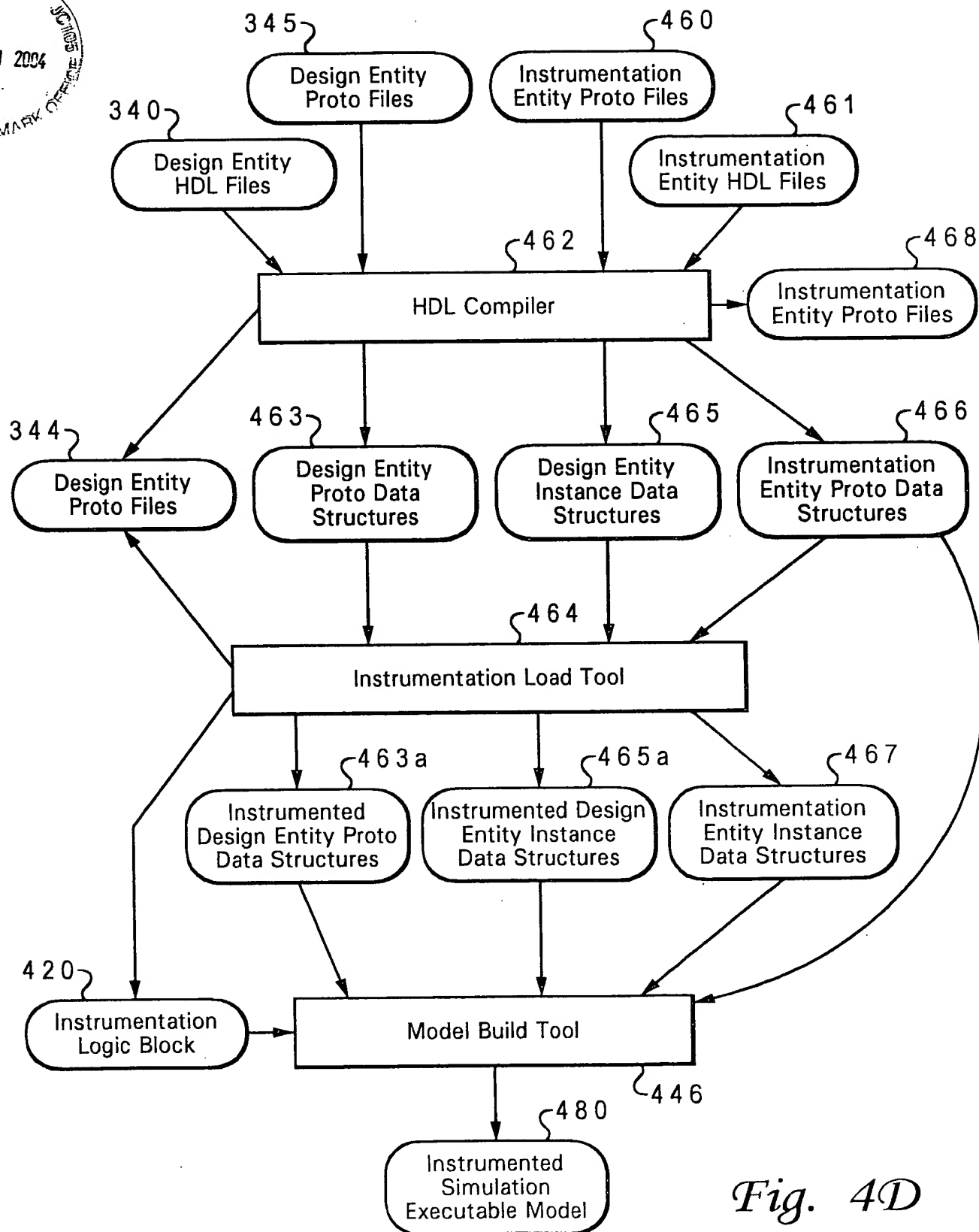


Fig. 4D



11/26

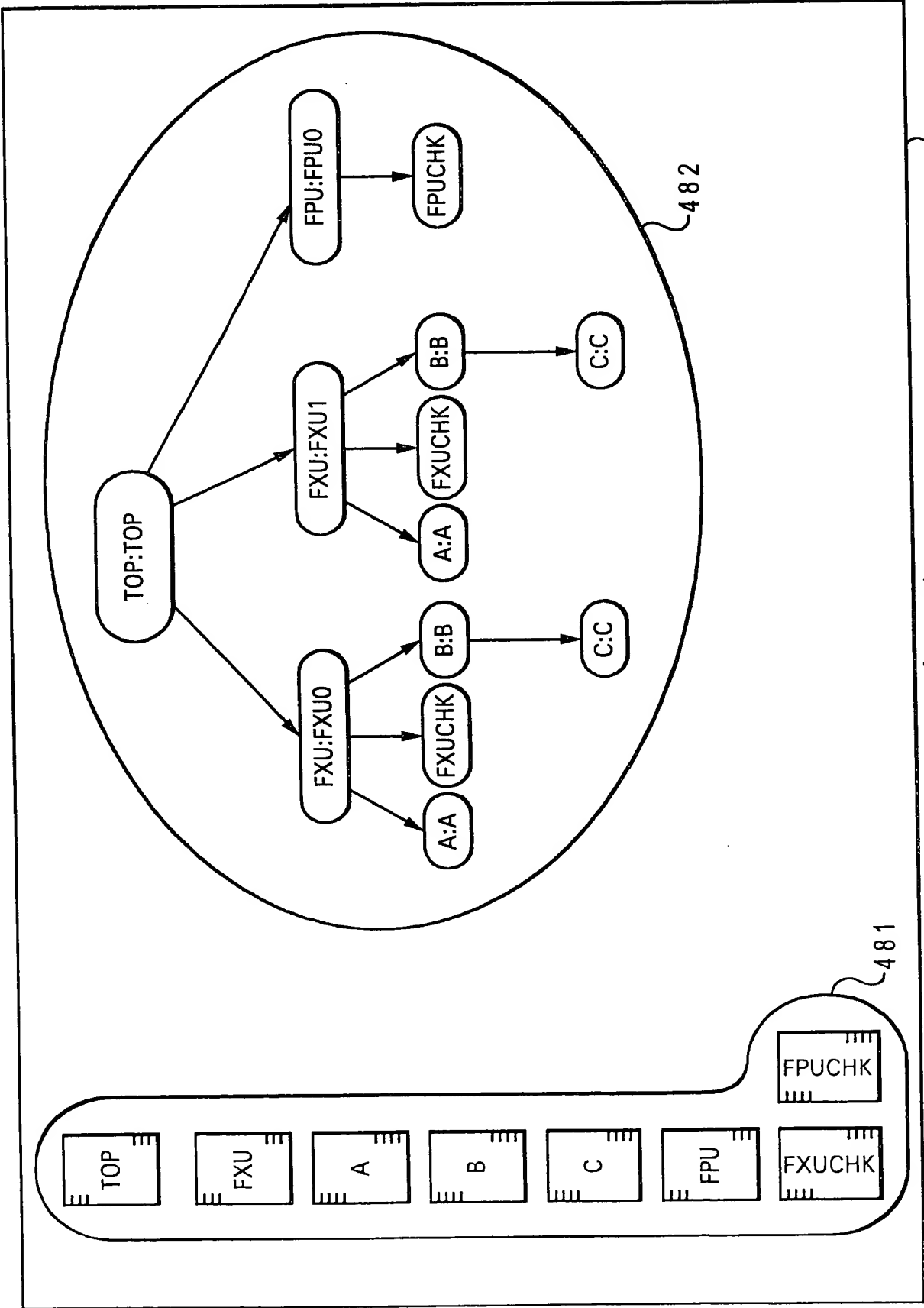


Fig. 4E



12/26

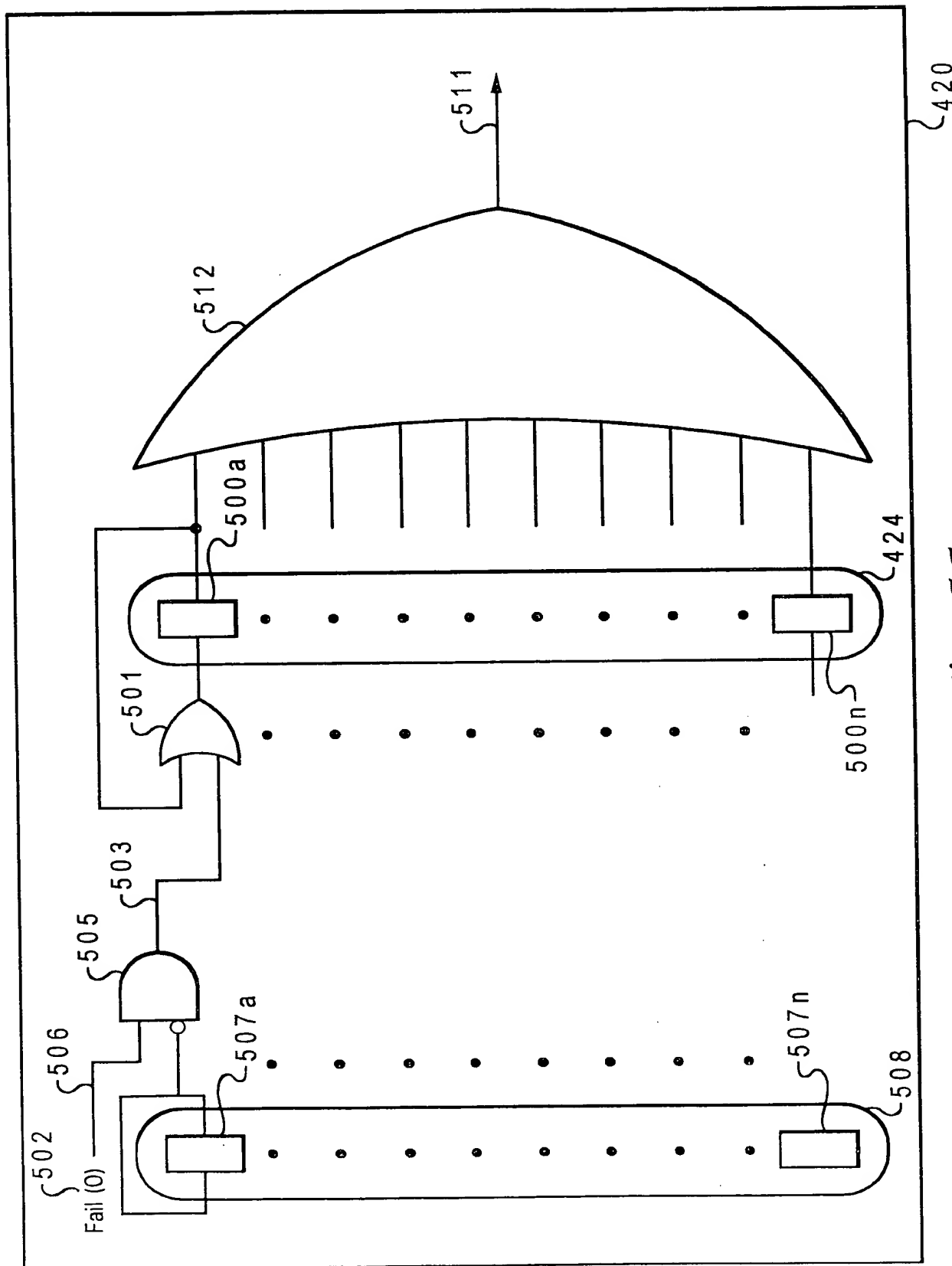


Fig. 5A

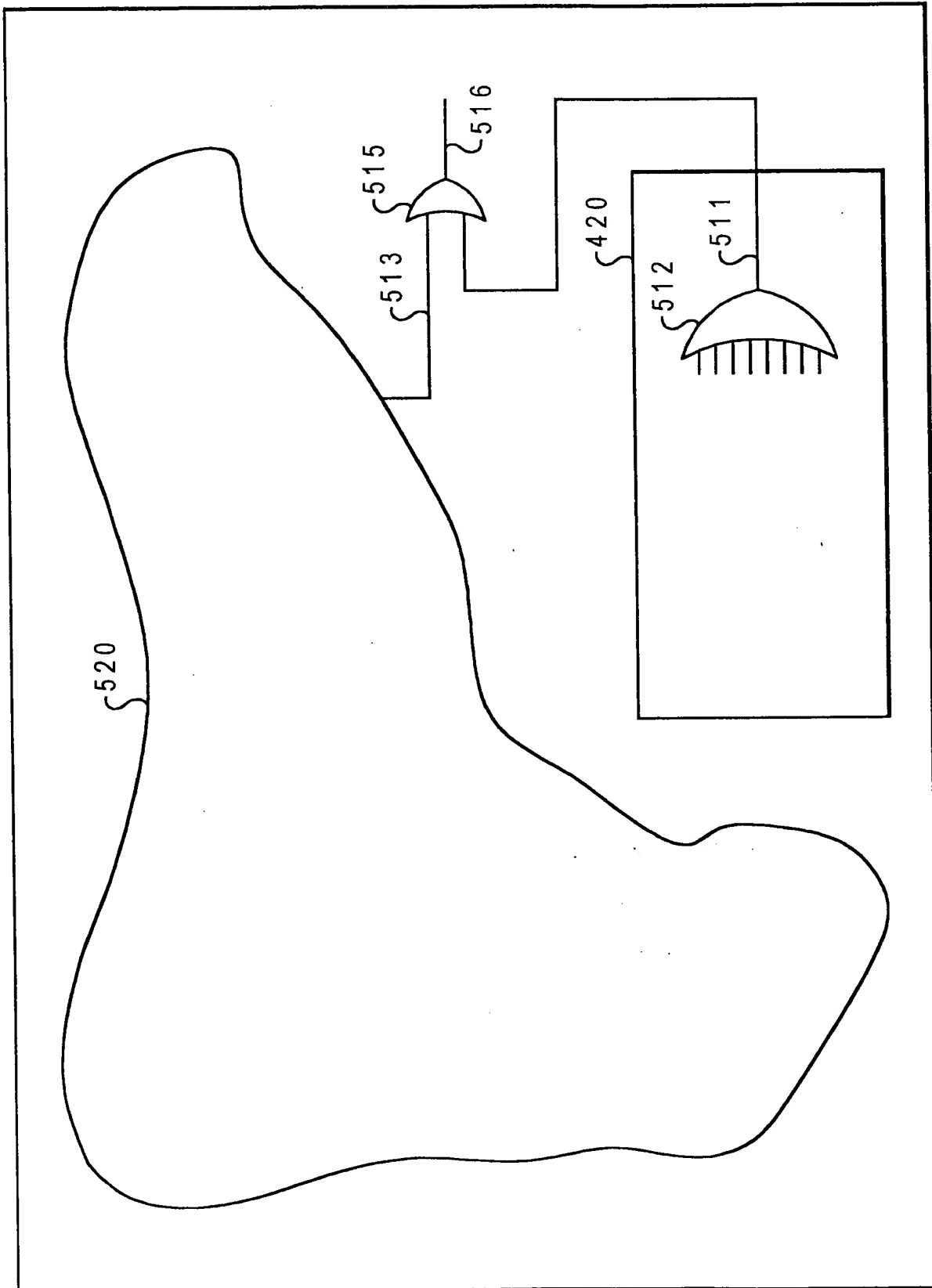
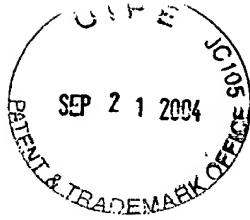


Fig. 5B

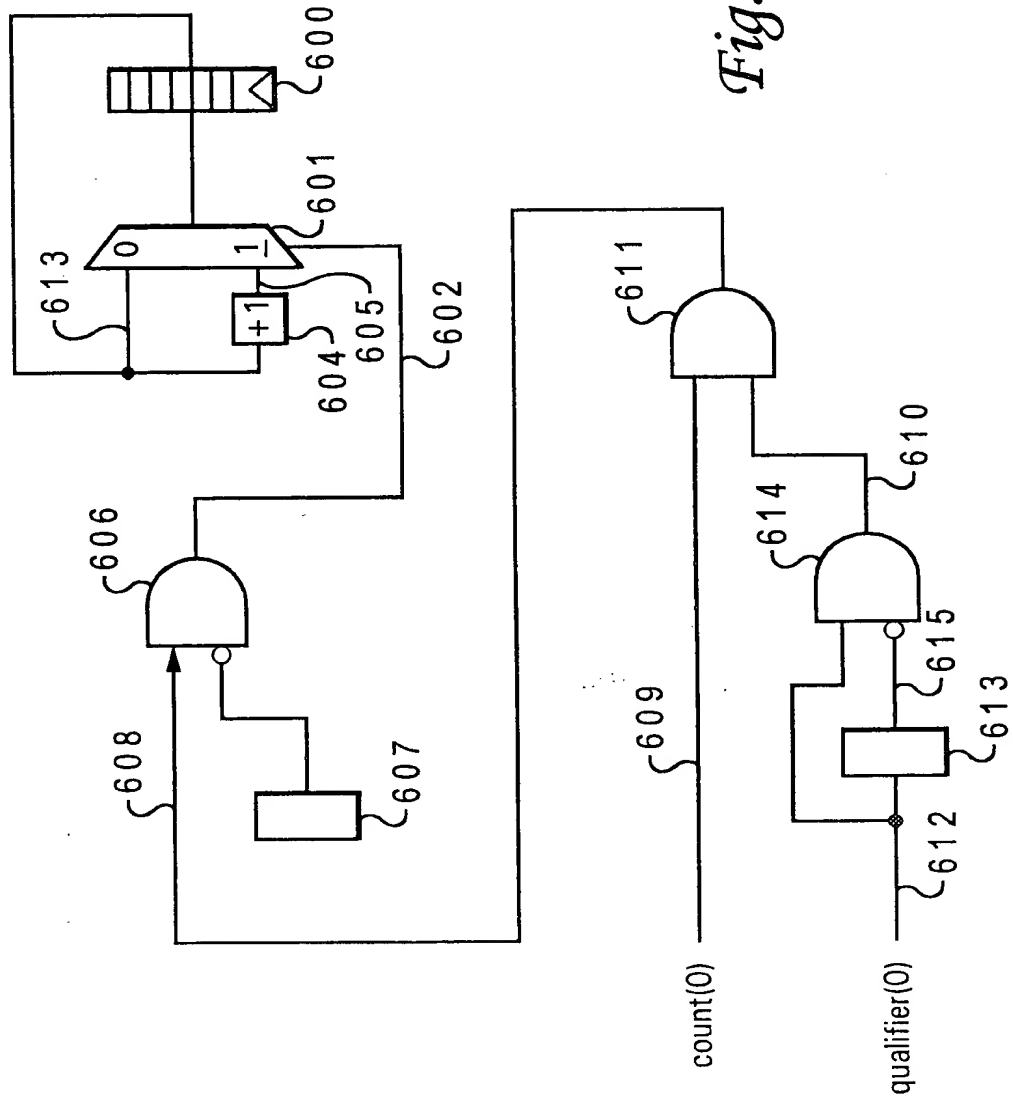


Fig. 6A



15/26

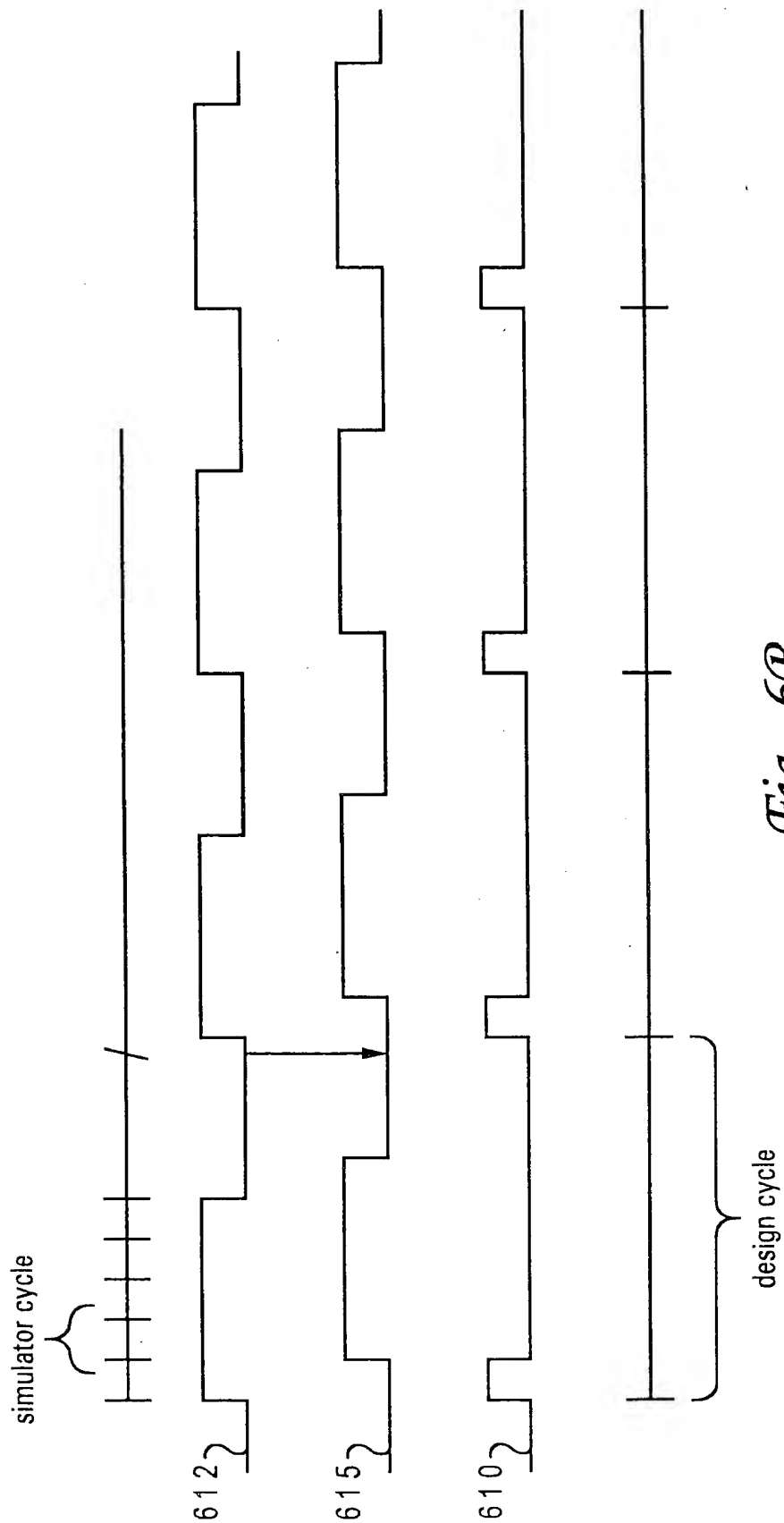


Fig. 6B

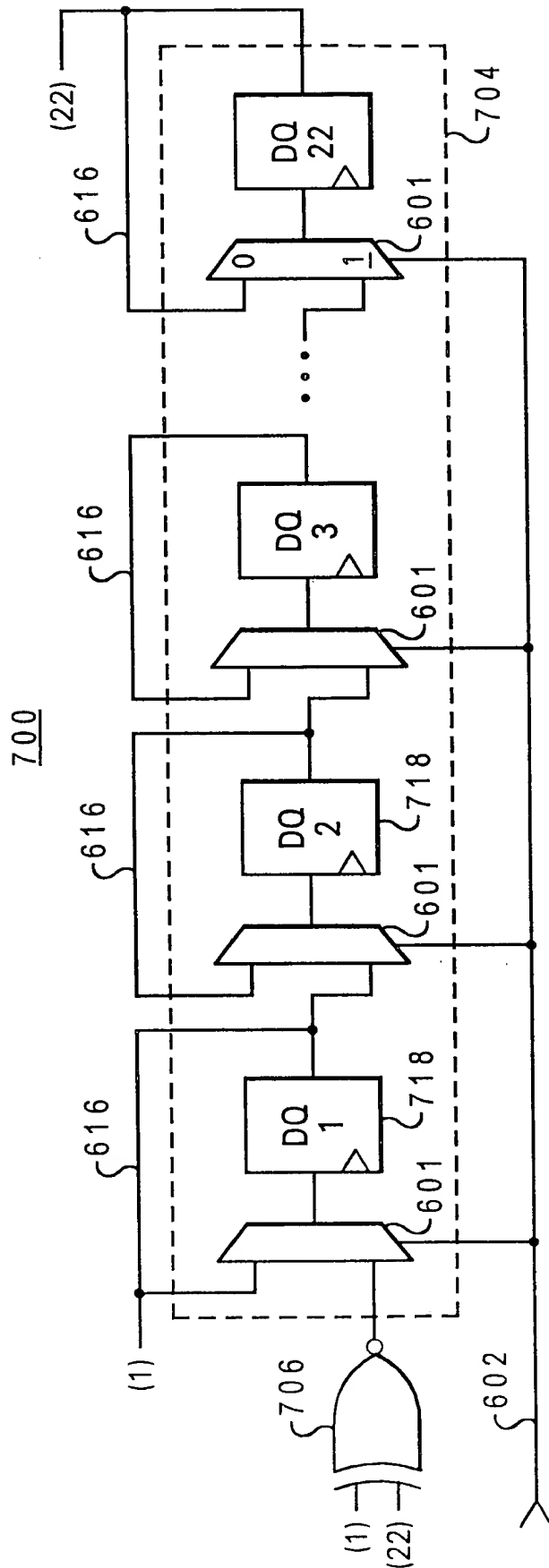
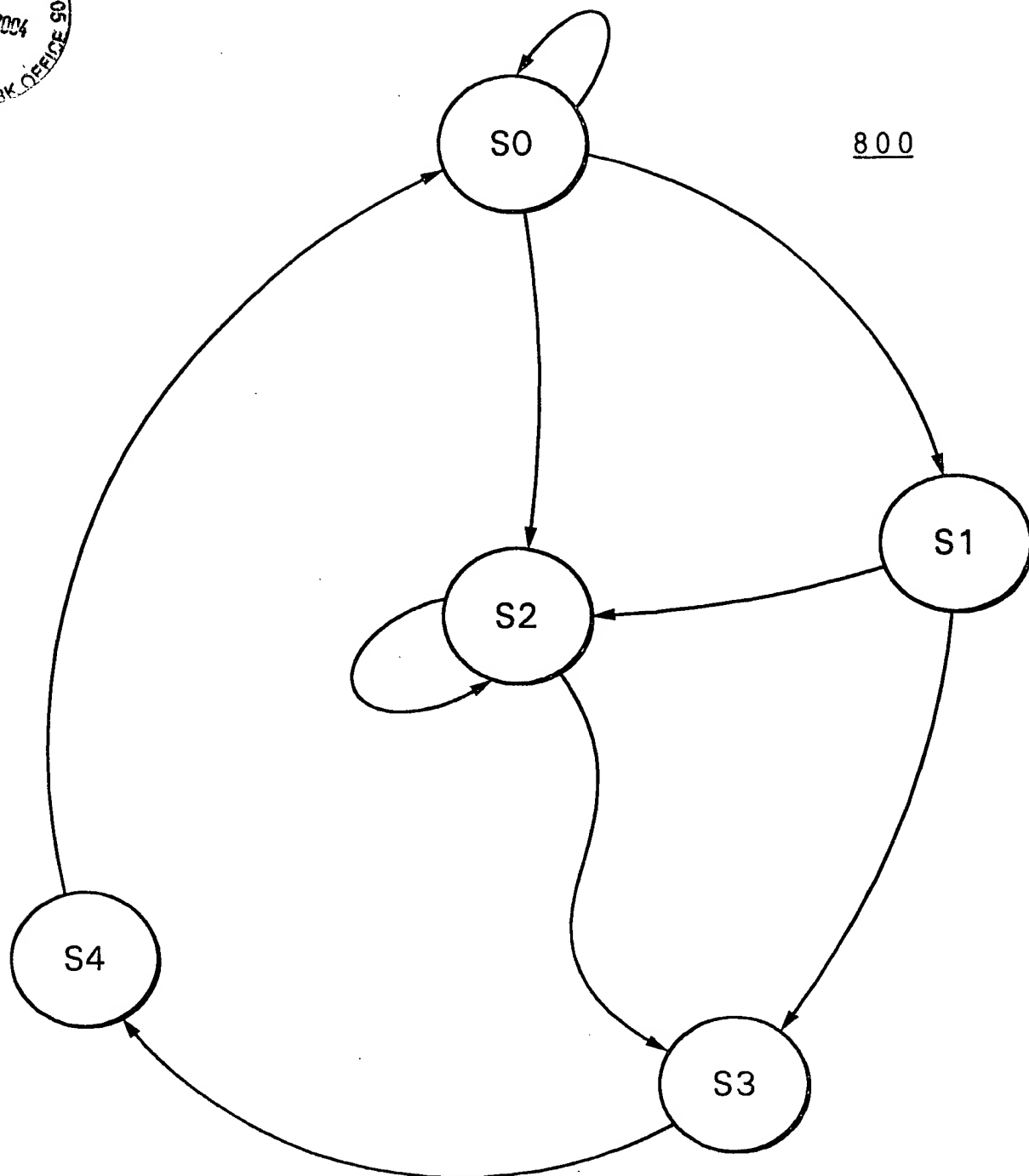


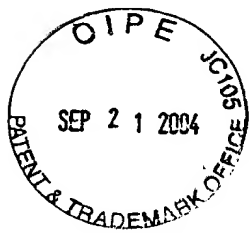
Fig. 7

17/26



800

Fig. 8A
Prior Art



entity FSM : FSM

850

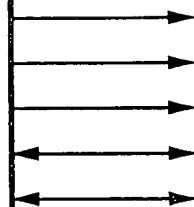
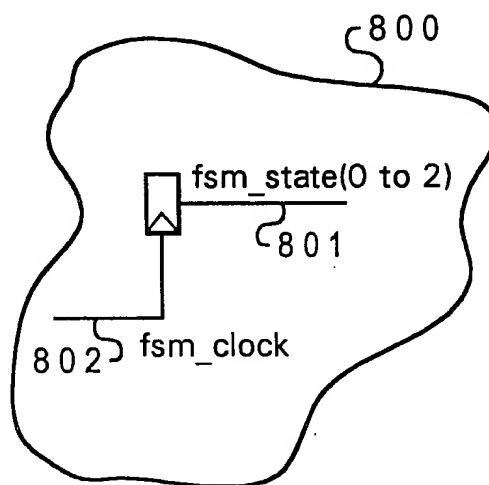
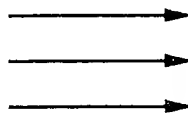


Fig. 8B
Prior Art

Fig. 8C

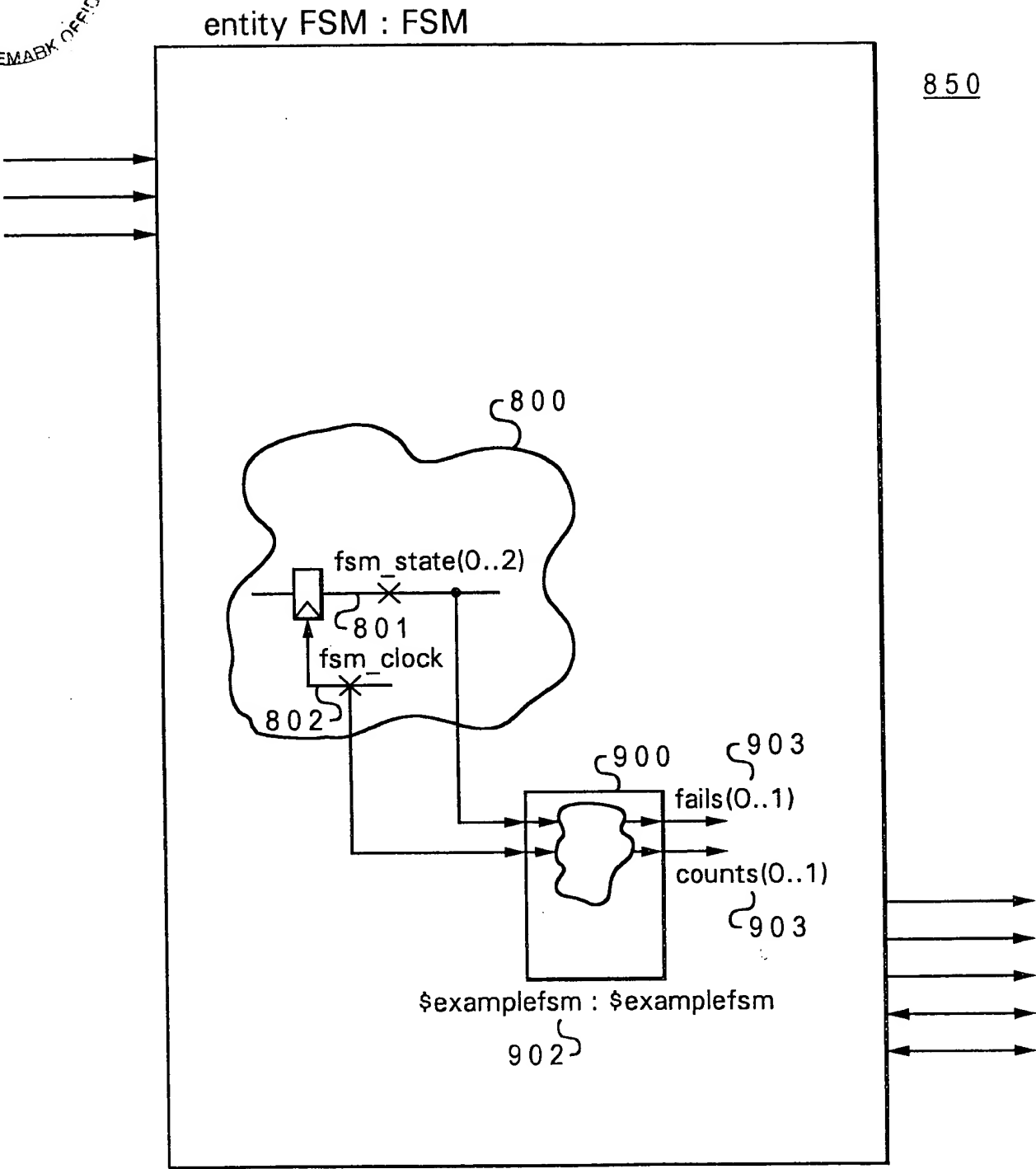
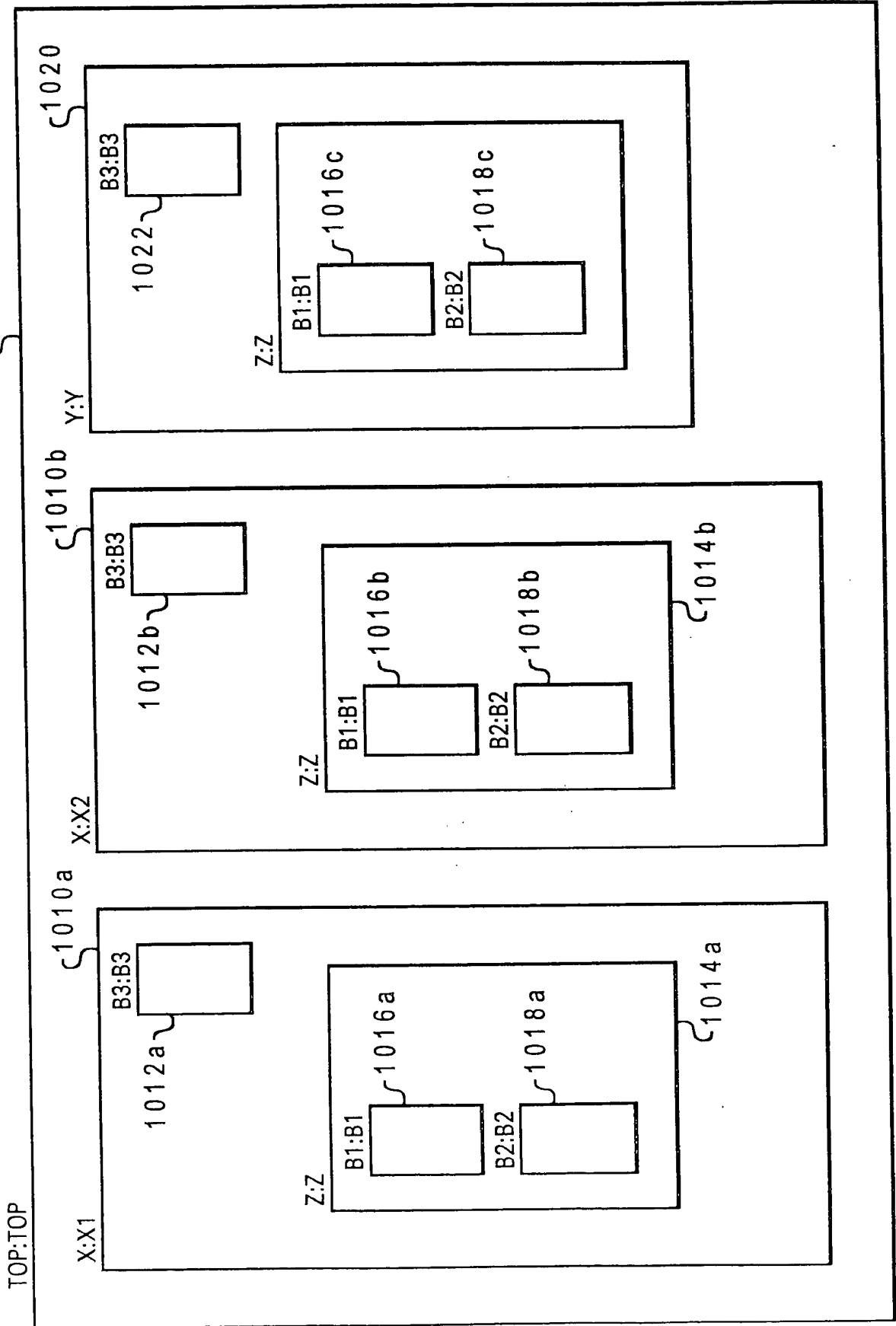


Fig. 9



Fig. 10A

1000



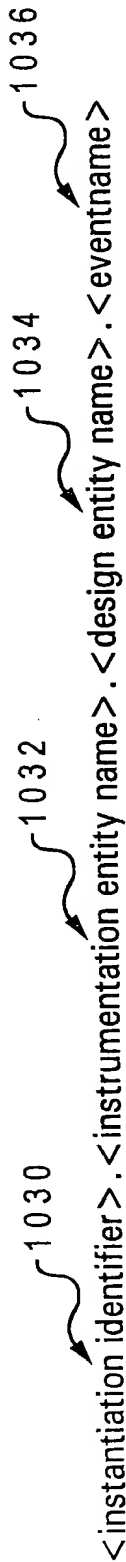


Fig. 10B

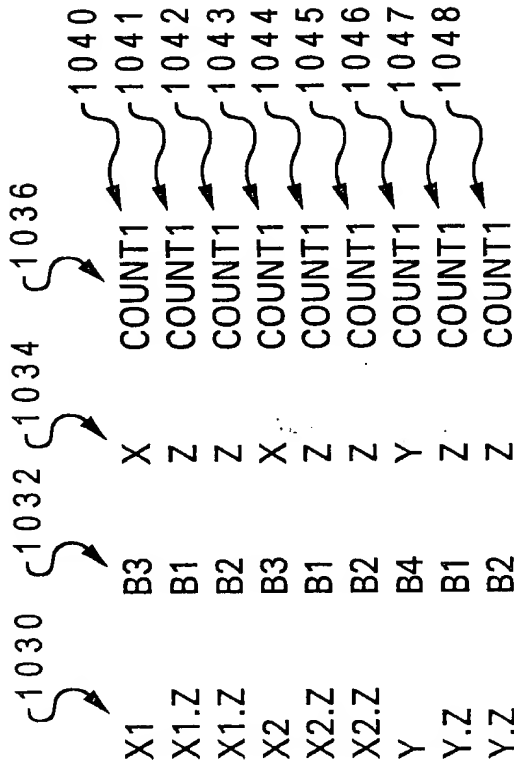


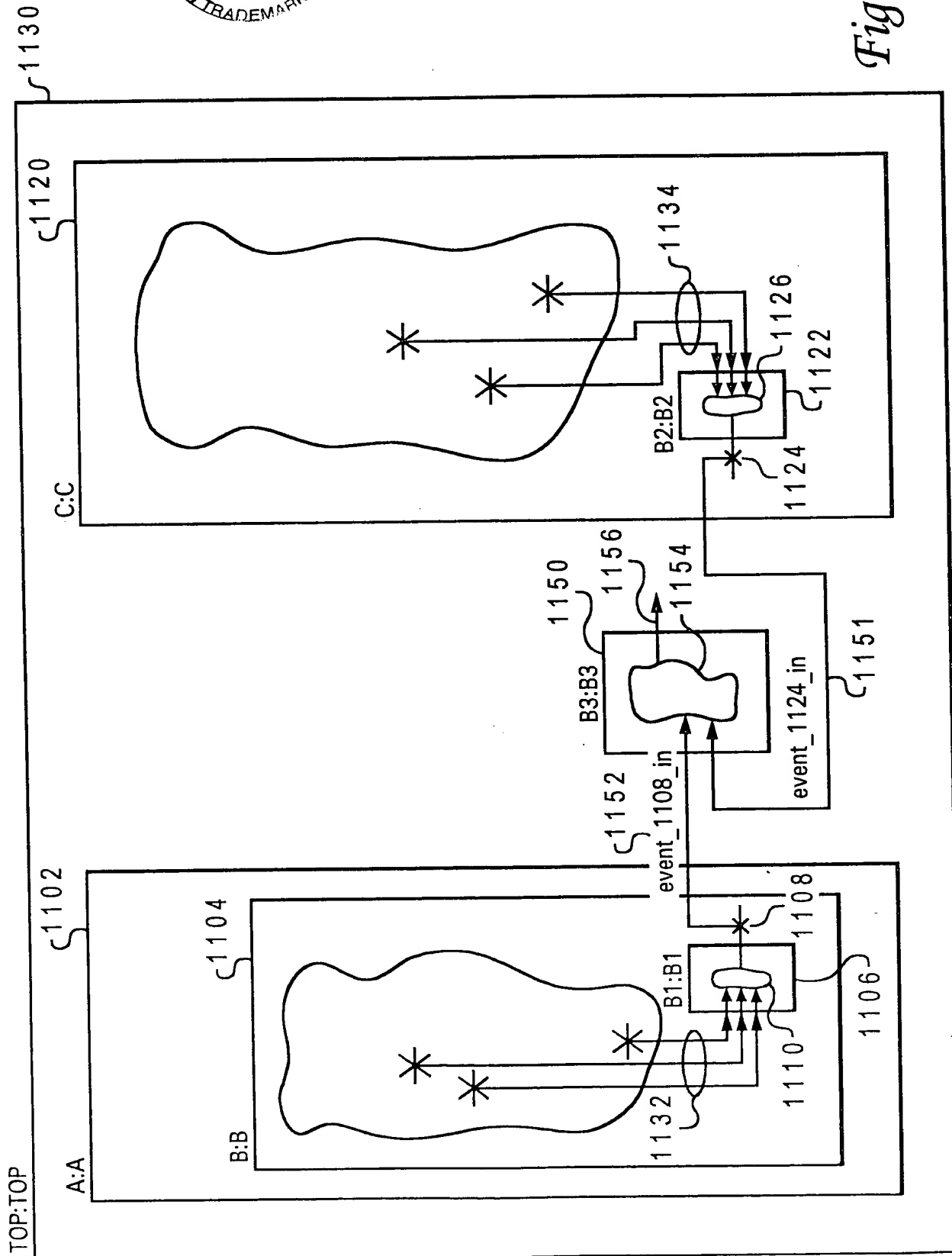
Fig. 10C



Fig. 10D



Fig. 11A





```

--!! Inputs
--!! event_1108_in <= C.[B2.count.event_1108];
--!! event_1124_in <= A.B.[B1.count.event_1124];
--!! End Inputs

```

Diagrammatic annotations for Fig. 11B:

- 1163 is above the first curly brace (under "Inputs").
- 1165 is above the second curly brace (under "event_1108_in").
- 1161 is above the third curly brace (under "event_1124_in").
- 1162 is below the third curly brace.
- 1164 is below the first curly brace.
- 1166 is below the second curly brace.

Fig. 11B

```

--!! Inputs
--!! event_1108_in <= C.[count.event_1108];
--!! event_1124_in <= B.[count.event_1124];
--!! End Inputs

```

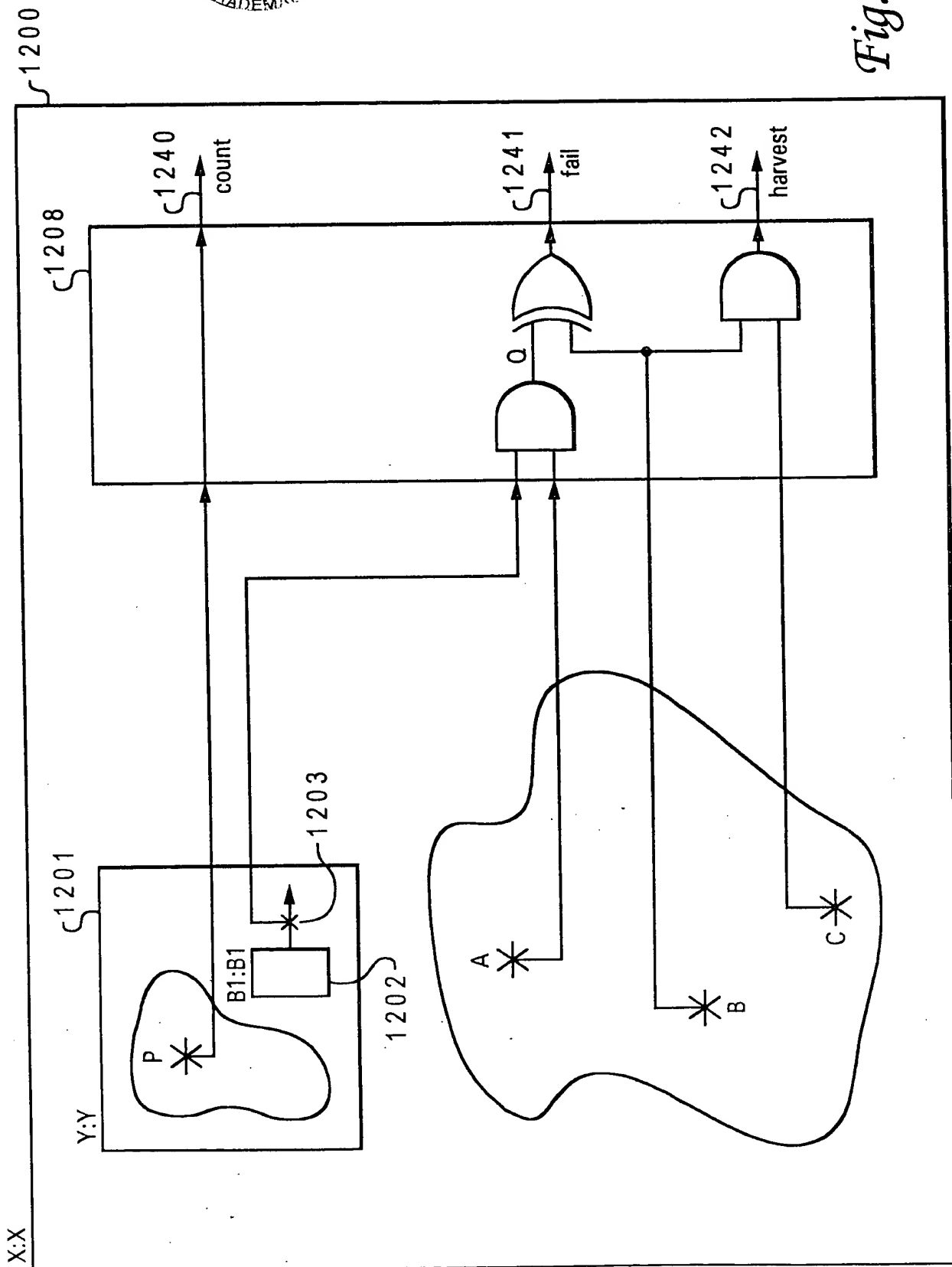
Diagrammatic annotations for Fig. 11C:

- 1171 is above the first curly brace (under "event_1108_in").
- 1172 is below the first curly brace.
- 1172 is below the second curly brace (under "event_1124_in").

Fig. 11C



Fig. 12A





```

ENTITY X IS
    PORT(
        :
        :
        :
    );

    ARCHITECTURE example of X IS
    BEGIN
        .
        .
        .
        ... HDL code for X ...
        .
        .
        .

1221 { Y:Y
      PORT MAP(
        :
        :
        );

1222 { A <= ....
      B <= ....
      C <= ....

1223 { --!! [count, countname0, clock] <= Y.P; 1230
      --!! Q <= Y. [B1.count.count1] AND A; 1232
      --!! [fail, failname0, "fail msg"] <= Q XOR B; 1234
      --!! [harvest, harvestname0, "harvest msg"] <= B AND C;
      END;
      1236
    
```

1220

Fig. 12B